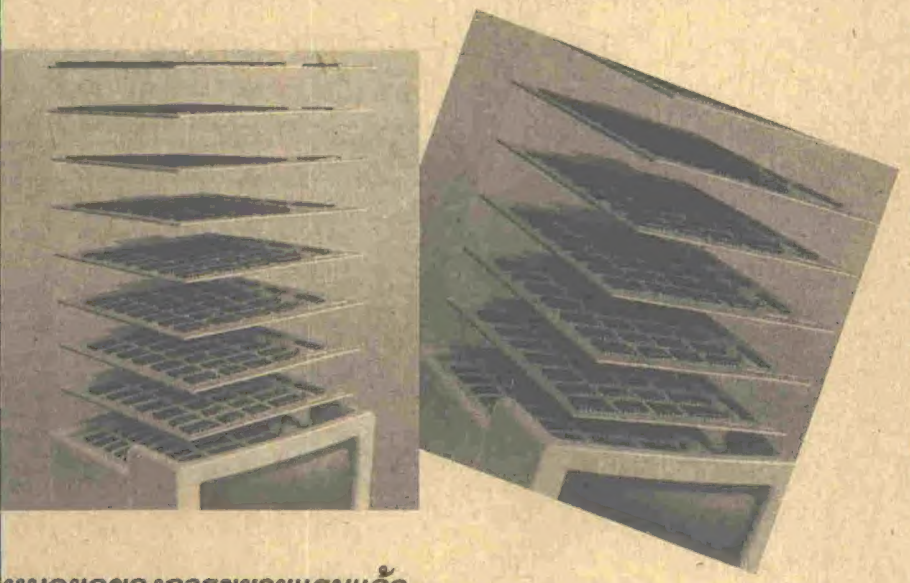
เรื่องน่าอ่าน

ประดิษฐ์ ภิญโญภาศกุล

ยกเครื่องเรื่อง หน่วยความจำเทียม



หมดยุคของการขยายแรมแล้ว เพราะเราสามารถนำหน่วยความจำเทียมมาใช้รันบนโปรแกรมใหญ่ ๆ ได้

หน่วยความจำเทียม (Virtual Memory) เป็นสิ่งที่จะช่วยให้คอมพิวเตอร์สามารถรันโปรแกรมที่มีขนาดใหญ่กว่าหน่วยความจำหลักที่มีอยู่จริง (RAM) ได้ หน่วยความจำเทียมถูกนำมาใช้เป็นครั้งแรกในระบบ Atlas ที่มหาวิทยาลัยแมนเชสเตอร์ของอังกฤษในปี พ.ศ. 2493 ปัจจุบันหน่วยความจำเทียมมีใช้กันทั่วไปในเครื่องทุกระดับ ไม่ว่าจะเป็นเมนเฟรม มินิ เวอร์กสเตชัน รวมทั้งไมโครคอมพิวเตอร์ด้วย

ลักษณะเด่นของหน่วยความจำเทียมก็คือ มีรายละเอียดที่เราต้องจัดการน้อยในการเก็บโปรแกรมไว้ในหน่วยความจำ หากเทียบกับโปรแกรมโอเวอร์เลย์ที่เราจำเป็นจะต้องกำหนดลงไปอย่างชัดเจนว่า ส่วนไหนของโปรแกรมที่จะถูกเก็บไว้ในหน่วยความจำอย่างต่อเนื่อง และส่วนไหนที่ถูกแยกออกมาเก็บในดิสก์ เมื่อมีการปรับปรุงโปรแกรมเป็นเวอร์ชันใหม่ เราจำเป็นจะต้องพิจารณาในส่วนนี้อีก แต่สำหรับหน่วยความจำเทียมแล้วเราไม่ต้องเป็นกังวลในส่วนนี้เลย

นอกจากนั้นแล้วในระบบหน่วยความจำเทียมเราสามารถจะสร้างโปรแกรมที่มีขนาดหลายเมกะไบต์โดยไม่ต้องสนใจว่าหน่วยความจำที่ใช้อยู่มีขนาดเท่าใด เพียงแต่ถ้าขนาดหน่วยความจำยิ่งมากโปรแกรมก็จะยิ่งรันได้เร็วขึ้น จนทำให้โปรแกรมแทบทุกโปรแกรมสามารถรันได้ทันที และไม่ต้องอาศัยโปรแกรมอื่นมาช่วยเลย

หลักการของระบบหน่วยความจำเทียมที่ทำให้ตัวมันเองมีความคล่องตัวในการใช้งานเป็นอย่างไร เราลองมาดูหลักการของโปรแกรมโอเวอร์เลย์กันก่อน ระหว่างการประมวลผลของโปรแกรมแบบโอเวอร์เลย์ บางส่วนของโปรแกรมจะถูกโหลดไปเก็บไว้ในหน่วยความจำ ส่วนที่เหลือจะเก็บไว้ในดิสก์ ตราบใดที่ส่วนของโปรแกรมที่ถูกอ้างถึงมีอยู่ในหน่วยความจำโปรแกรมจะยังดำเนินต่อไป จนกระทั่งโปรแกรมอ้างถึงสิ่งที่ไม่อยู่ในหน่วยความจำแล้ว ระบบจัดการของเครื่องจะทำการขัดจังหวะและสับเปลี่ยนส่วนที่ถูกอ้างถึงจากดิสก์เข้าไปเก็บในหน่วยความจำแทน

ในระบบหน่วยความจำเทียม โปรแกรมจะถูกแบ่งเป็นส่วนย่อย ๆ ซึ่งส่วนย่อยของโปรแกรมเหล่านี้สามารถสับเปลี่ยนที่อยู่ระหว่างหน่วยความจำกับดิสก์เมื่อใดก็ได้ และระบบจัดการหน่วยความจำเทียมจะจดจำตำแหน่งของแต่ละส่วนเอาไว้ ถ้าแต่ละส่วนมี

ขนาดต่างกันเราจะเรียกส่วนย่อยเหล่านั้นว่าเซกเมนต์ (Segments) ซึ่งจะมีอย่างน้อยหนึ่งโค้ดเซกเมนต์กับหนึ่งดาต้าเซกเมนต์ ถ้าแต่ละส่วนมีขนาดเท่ากันเราจะเรียกว่าเพจ (pages) ไม่ว่าส่วนย่อยเป็นแบบใดจะต้องมีระบบจัดการหน่วยความจำเทียมเข้ามาช่วยในการโอนย้ายส่วนที่ถูกอ้างถึงระหว่างดิสก์กับหน่วยความจำ เป็นที่ยอมรับกันทั่วไปในปัจจุบันว่าการแบ่งส่วนย่อยของโปรแกรมเป็นเพจมีข้อดีมากกว่าการแบ่งเป็นเซกเมนต์ ระบบหน่วยความจำเทียมที่นิยมกันจึงเป็นแบบ demand-paged VM ซึ่งต่างไปจากระบบ anticipatory paging เดิมตรงที่ระบบจัดการหน่วยความจำเทียมจะพยายามทำนายไว้ล่วงหน้าว่าเพจไหนที่โปรแกรมจะอ้างถึงต่อไป

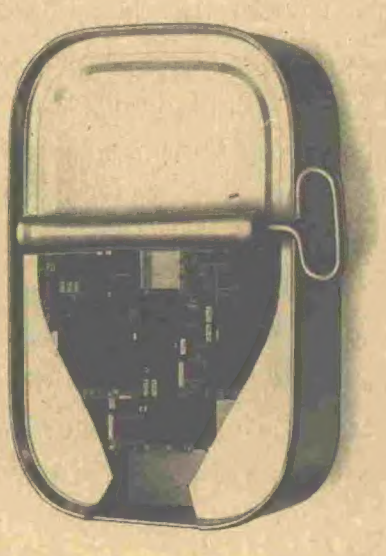
ระบบจัดการหน่วยความจำเทียมเป็นระบบจัดการย่อยที่ยังคงรักษาโครงสร้างข้อมูลเดิมซึ่งจำเป็นในการแปลงแอดเดรสเทียมให้เป็นแอดเดรสจริงที่จะนำส่วนของโปรแกรมที่ถูกอ้างถึงไปเก็บไว้ อีกทั้งยังสามารถใช้ได้กับทั้งระบบจัดการเดี่ยว (single-tasking) และหลายระบบจัดการ (multi-tasking) โดยการจัดแบ่งแอดเดรสเทียมออกเป็นส่วน ๆ สำหรับแต่ละระบบจัดการ

หน่วยความจำเทียมบน 386 และ 486

ซีพียู 32 บิตของอินเทลมีความสามารถในการอ้างถึงหน่วยความจำเกินกว่าที่มีอยู่จริงได้ อันเป็นผลมาจากระบบฮาร์ดแวร์ที่ได้เตรียมการไว้ให้ใช้กับระบบจัดการหน่วยความจำเทียมนั่นเอง นอกจากนี้ซอฟต์แวร์ที่ช่วยสนับสนุนก็เป็นสิ่งจำเป็นสำหรับการใช้ระบบหน่วยความจำเทียมบนซีพียู 80386 และ 80486 ซึ่งในปัจจุบันเราจะพบระบบจัดการย่อยหน่วยความจำเทียมในเวอร์ชันพิเศษของยูนิกซ์และซีนิกซ์ รวมทั้งซอฟต์แวร์ของ Phar Lap ตลอดจนโอเอสทูเวอร์ชันที่ออกใหม่ก็สนับสนุนการใช้หน่วยความจำเทียมเช่นกัน

พื้นฐานของระบบฮาร์ดแวร์

หลังจากซีพียู 8086 และ 8088 พัฒนาขึ้นมาเป็น 80286 อินเทลได้ออกแบบให้ซีพียู 80286 สามารถใช้กับระบบ segmented VM ได้ โดยแบ่งการทำงานออกเป็น 2 โหมดคือ real mode ใช้รันดอส และ protected mode ซึ่งจะสนับสนุนการใช้ระบบหน่วยความจำเทียมสำหรับซีพียู 80386 ที่ออกตามมาภายหลัง นอกจากอินเทลจะพัฒนา 80386 ให้เป็นซีพียู 32 บิตแล้ว อินเทลยังได้เพิ่มระบบฮาร์ดแวร์ที่สนับสนุนการใช้ระบบ paged VM อีกด้วย ถึงตอนนี้ไม่ว่าจะเป็นซีพียู 80286, 80386 หรือแม้แต่ 80486 ต่างก็มีระบบฮาร์ดแวร์ที่สามารถใช้งานระบบหน่วยความจำเทียมได้ อาจจะโชคร้ายอยู่นิดตรงที่ว่าดอสจำเป็นจะต้องรันใน real mode แต่หน่วยความจำเทียมสามารถใช้ได้ใน protected mode เท่านั้น อย่างไรก็ตาม เมื่อซีพียู 80286 และ 80386 ออกมาไม่นานนัก ผู้พัฒนาระบบยูนิกซ์และซีนิกซ์ได้ออกเวอร์ชันใหม่ที่สนับสนุน segmented VM บน 80286 และ paged VM บน 80386 ขึ้น สำหรับโอเอสทูก็สนับสนุน segmented VM เหมือนกัน ระบบจัดการเหล่านี้ต่างช่วยให้เครื่องคอมพิวเตอร์สามารถรันโปรแกรมที่มีขนาดใหญ่ขึ้นได้ แต่ยังรันโปรแกรมดอสที่มีขนาดใหญ่กว่า 640 กิโลไบต์ไม่ได้



เข้าเรื่องสถาปัตยกรรมของหน่วยความจำ

ซีพียู 80386 และ 80486 มีระบบฮาร์ดแวร์ที่สนับสนุนหน่วยความจำเทียมทั้งสองแบบ เมื่อทำงานใน protected mode หน่วยความจำจึงจำเป็นต้องมีสถาปัตยกรรมผสมระหว่าง paged VM และ segmented VM

Segmented VM

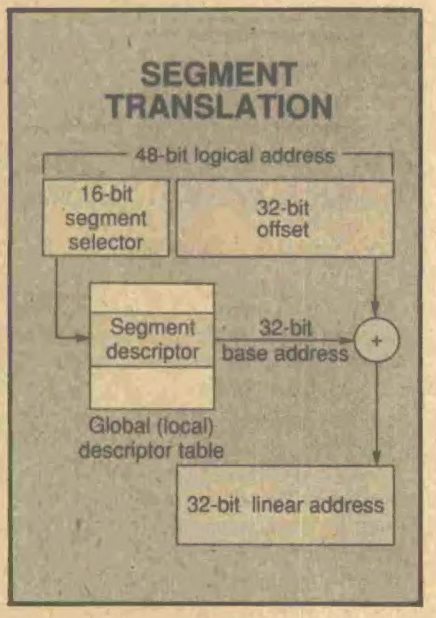
โปรแกรมสำหรับซีพียู 80386 และ 80486 เมื่อรันในระบบนี้จะถูกแบ่งออกเป็นเซกเมนต์อย่างน้อยหนึ่งโค้ดเซกเมนต์กับอีกหนึ่งดาต้าเซกเมนต์ โดยที่โค้ดเซกเมนต์จะเปลี่ยนแปลงแก้ไขไม่ได้ ในขณะที่ส่วนของดาต้าเซกเมนต์ไม่สามารถทำงานได้ ข้อจำกัดของเซกเมนต์ทั้งสองประเภทนี้ทำให้สามารถจะตรวจหาข้อผิดพลาดทั่ว ๆ ไปของโปรแกรมได้

เซกเมนต์ของโปรแกรมบนซีพียู 80386 และ 80486 ต่างมีแอดเดรสเริ่มต้นและสิ้นสุด มีขนาดของแต่ละเซกเมนต์จำกัด และมีข้อมูลที่จะจำกัด

ประเภทและลักษณะเฉพาะของเซกเมนต์ ว่าเซกเมนต์นั้น ๆ เป็นโค้ดเซกเมนต์ หรือดาต้าเซกเมนต์ อ่านได้อย่างเดียว หรือทั้งอ่านและเขียน ข้อมูลต่าง ๆ เหล่านี้เรียกว่า segment descriptor ซึ่งเก็บไว้ในตาราง (descriptor table) สามารถแบ่งแยกย่อยได้เป็น GDT (Global Descriptor Table) และ LDT (Local Descriptor Table)

เมื่อโปรแกรมหนึ่ง ๆ อ้างถึงแอดเดรสในหน่วยความจำมักจะหมายถึงตำแหน่งภายในเซกเมนต์ที่โปรแกรมกำลังใช้งานอยู่ ในซีพียู 80386 และ 80486 จะมีเซกเมนต์รีจิสเตอร์ซึ่งผ่านการให้ค่าเมื่อโปรแกรมถูกโหลดลงในหน่วยความจำ ค่าที่ให้กับเซกเมนต์-รีจิสเตอร์เราเรียกว่า segment selectors จะเป็นตัวกำหนดว่าจะใช้ตาราง GDT หรือ LDT

โปรแกรมสามารถอ้างถึงหน่วยความจำในรูปของออฟเซตภายในเซกเมนต์ที่ถูกเลือกได้ และเมื่อจะอ้างถึงเซกเมนต์อื่น ๆ ผู้ใช้เพียงแต่โหลดเฉพาะเซกเมนต์รีจิสเตอร์อีกครั้งเท่านั้น หากเราใช้ภาษาระดับสูงในการเขียนโปรแกรมก็ไม่ต้องจัดการในส่วนนี้ เพราะคอมพายเลอร์จะเป็นผู้จัดการให้เอง



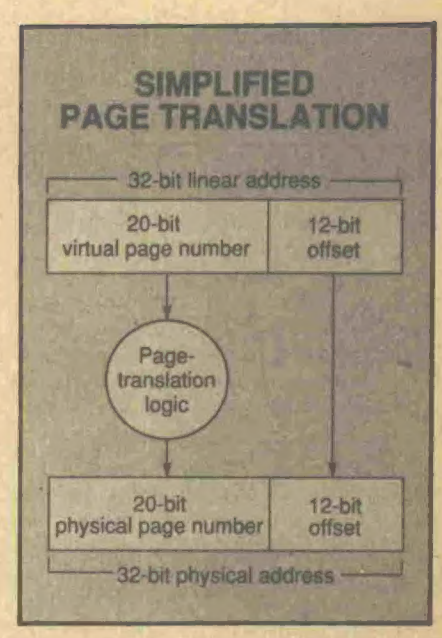
รูป 1 แสดงการแปลงแอดเดรสของระบบ segmented VM

segment selector ตัวหนึ่งร่วมกับอีกออฟเซตหนึ่งของเซกเมนต์จะใช้อ้างถึงลอจิคัลแอดเดรส (logical address) หรือที่เรียกกันเป็นส่วนใหญ่ว่าแอดเดรสเทียม (virtual address) เนื่องจากเป็นเพียงแอดเดรสที่โปรแกรมมองเห็น ลอจิคัลแอดเดรสจะถูกแปลงโดยหน่วยแปลงแอดเดรสของโปรเซสเซอร์ให้เป็นลิเนียร์แอดเดรส (linear address) ซึ่งจะประกอบด้วยแอดเดรสเริ่มต้นของเซกเมนต์กับออฟเซตภายในเซกเมนต์นั้น ดังรูปที่ 1

Paged VM

ซีพียู 80386 และ 80486 มีระบบแอดเดรสบัสทั้งภายในและภายนอกเป็นแบบ 32 บิต ขนาดสูงสุดของหน่วยความจำหลักและหน่วยความจำเทียมจึงเป็น 4 กิกะไบต์ (2^{32} ไบต์) หน่วยความจำหลักของระบบสามารถแบ่งออกเป็นเพจ มีขนาดสูงสุด 4 กิโลไบต์ เรียงกันเป็นแถวตรง ดังนั้นเราจึงสามารถจะมองหน่วยความจำหลักของคอมพิวเตอร์เสมือนกับอะเรย์ของเพจซึ่งเริ่มต้นจากแอดเดรสศูนย์ เพจจะถูกจัดการโดยระบบจัดการย่อยในหน่วยความจำเทียม ไม่ใช่ตัวโปรแกรม

เพจของหน่วยความจำหลักจะอ้างถึงเป็นเพจจริง (physical pages) หรืออีกชื่อหนึ่งว่าเพจเฟรม (page frames) ถ้าเป็นเครื่องไมโครคอมพิวเตอร์ที่มีหน่วยความจำหลักอยู่ 2 เมกะไบต์ จะมีเพจจริงได้ 512 เพจ แต่จะอ้างถึงทั้งเพจจริงและเพจเทียมรวมกันได้ถึง



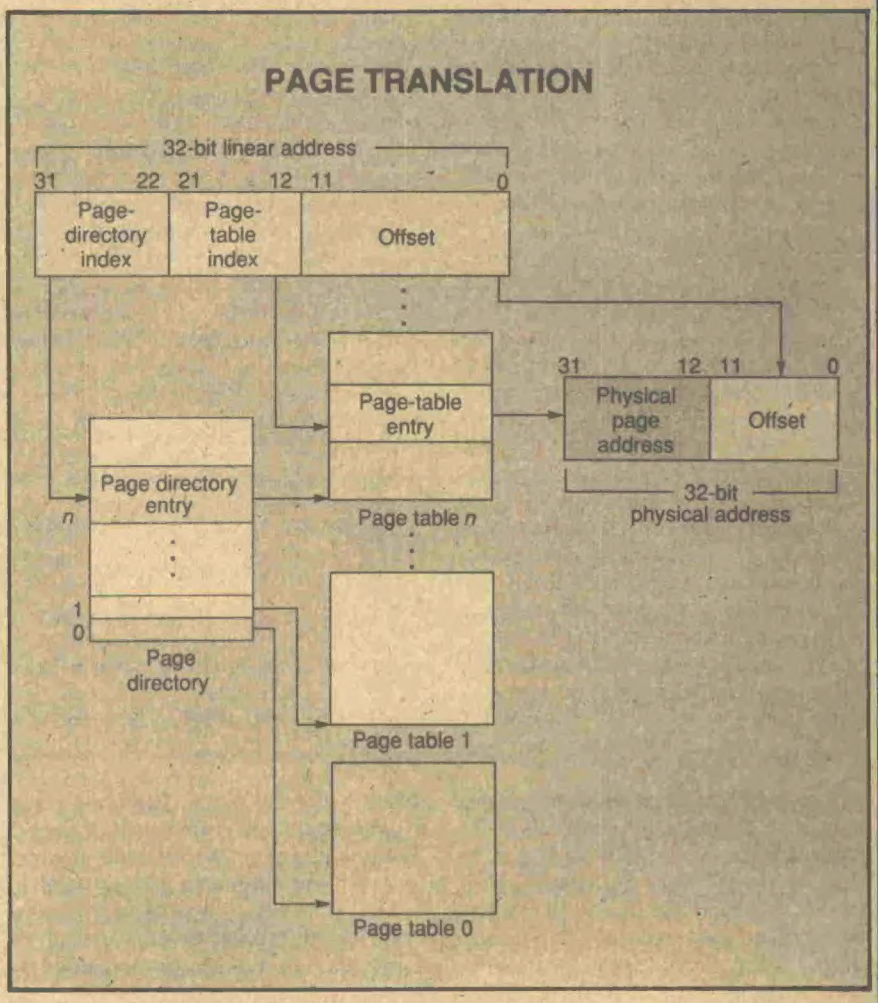
รูป 2 แสดงการแปลงแอดเดรสโดยย่อของระบบ paged VM

1,048,576 เพจหรือ 4 กิกะไบต์ ยิ่งถ้าไมโครคอมพิวเตอร์เครื่องนั้นใช้หลายระบบจัดการก็จะอ้างถึงแอดเดรสเทียมได้มากขึ้นอีก โดยการแบ่งตารางเพจออกเป็นส่วน ๆ สำหรับแต่ละระบบจัดการ ซึ่งในความเป็นจริงแล้วหน่วยความจำที่มีอยู่จริงมีขนาดเล็กกว่า 4 กิกะไบต์ที่อ้างถึงได้อยู่มาก อย่างไรก็ตาม ขนาดของหน่วยความจำเทียมจะถูกจำกัดโดยเนื้อที่ว่างของดิสก์ เพราะเพจเทียมจะถูกเก็บไว้ในหน่วยความจำหลักหรือไม่ก็ในดิสก์

ในการอ้างถึงตำแหน่งในเพจจะใช้ลิเนียร์แอดเดรสซึ่งแปลงให้เป็นแอดเดรสจริง (physical address) ของหน่วยความจำที่ติดตั้งอยู่บนบัสภายนอก ลิเนียร์แอดเดรสในระบบ paged VM ประกอบด้วยหมายเลขออฟเซตขนาด 12 บิต ทำให้แต่ละเพจมีขนาด 4,096 ไบต์ (2^{12} ไบต์) และหมายเลขเพจเทียมขนาด 20 บิต ดังรูปที่ 2

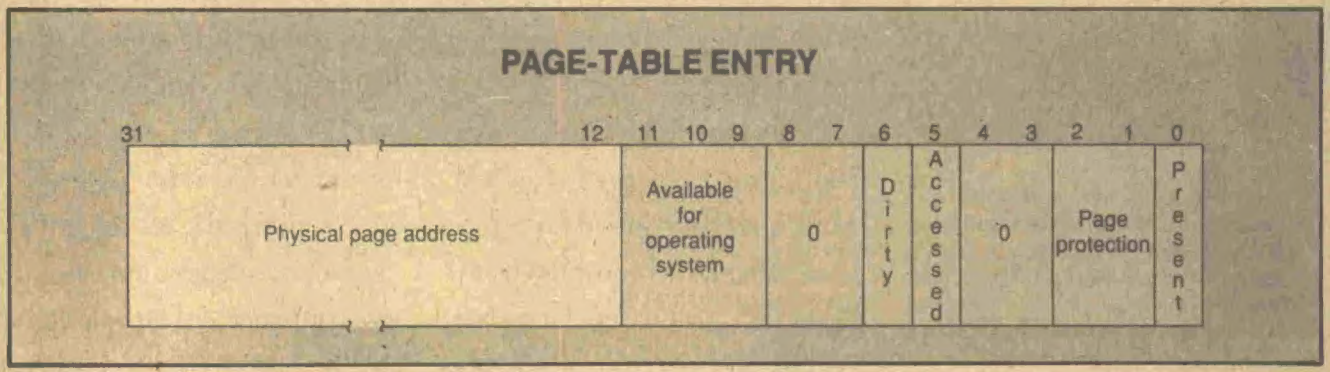
ระบบจัดการย่อยของหน่วยความจำเทียมจะแปลงหมายเลขเพจเทียมให้เป็นหมายเลขเพจจริง เมื่อรวมกันกับหมายเลขออฟเซตก็จะได้แอดเดรสจริง ตัวอย่างเช่น เพจเทียมหมายเลข 8 อาจจะมีลิเนียร์แอดเดรสเริ่มต้นที่ 32K และสิ้นสุดที่ 36K อาจจะตรงกับเพจจริงหมายเลข 1 ซึ่งมีแอดเดรสจริงเริ่มต้นที่ 4K และสิ้นสุดที่ 8K หากอ้างถึงลิเนียร์-แอดเดรสที่ 33K (00008400 ในระบบเลขฐานสิบหก) จะหมายถึงแอดเดรสจริงที่ 5K (00001400 ในระบบเลขฐานสิบหก) โดยการที่เพจเทียมหมายเลข 8 ถูกแปลงเป็นหมายเลขเพจจริงตรงกับหมายเลข 1

ในรูปที่ 3 แสดงการแปลงแอดเดรสในระบบ paged VM ในรายละเอียดที่มากขึ้น หมายเลขเพจเทียมขนาด 20 บิตจะประกอบด้วย 2 ส่วนที่มีขนาด 10 บิตเท่ากัน ได้แก่ ตัวชี้ไปยังสารบาญเพจ (page-directory index) และตัวชี้ไปยังตารางเพจ (page-table index) ในระบบ paged VM จะมีสารบาญเพจอยู่หนึ่งอันและมีหลายตารางเพจ สารบาญเพจและตารางเพจมีขนาด 4 กิโลไบต์ ซึ่งแต่ละอันจะบรรจุข้อมูลขนาด 32 บิตได้ 1,024 ชุด ข้อมูลในสารบาญเพจจะเป็นแอดเดรสของเพจจริง กล่าวโดยสรุปก็คือ การแปลงแอดเดรสในระบบ paged VM จะใช้ตัวชี้ไปยังสารบาญเพจจริงเพื่อให้ได้แอดเดรสของตารางเพจที่เหมาะสมมาค้นหาแอดเดรสของเพจจริง



รูป 3 แสดงการแปลงแอดเดรสในรายละเอียดของระบบ paged VM

ข้อมูลในตารางเพจสำหรับแต่ละเพจจริงจะแสดงได้ดังรูปที่ 4 ข้อมูลแต่ละชุดจะประกอบด้วยแอดเดรสของ



รูป 4 แสดงข้อมูลชุดหนึ่ง ๆ ในตารางเพจ

เพจจริงขนาด 20 บิต ที่เหลืออีก 12 บิต จะเผื่อไว้สำหรับข้อมูลเพิ่มเติมอื่น ๆ กล่าวคือ

- บิต 0 จะแสดงสถานะว่าเพจนั้นอยู่ในหน่วยความจำหรืออยู่ในดิสก์
- บิต 1 และบิต 2 แสดงสถานะว่าเพจนั้นอ่านได้อย่างเดียว อ่านและเขียนได้ หรือสงวนไว้ใช้เฉพาะสำหรับระบบจัดการหน่วยความจำเทียม
- บิต 5 แสดงสถานะว่าเพจนั้นถูกใช้อยู่หรือไม่
- บิต 6 แสดงสถานะว่าเพจนั้นถูกเปลี่ยนแปลงแก้ไขหรือไม่
- บิต 9-11 สำรองไว้ให้ระบบจัดการหน่วยความจำเทียมได้ใช้งาน

ส่วนอีก 4 บิตที่ไม่ได้กล่าวถึงจะสำรองไว้สำหรับซีพียู 80386 แต่ในซีพียู 80486 บิตที่ 3 และ 4 จะใช้ในการควบคุมหน่วยความจำแคช (cache memory) ในระดับเพจ ถ้าเพจใดไม่อยู่ในหน่วยความจำ (บิต 0 มีค่า 0) ระบบจัดการย่อยของหน่วยความจำเทียมจะสามารถใช้งานบิตที่เหลืออีก 31 บิตได้

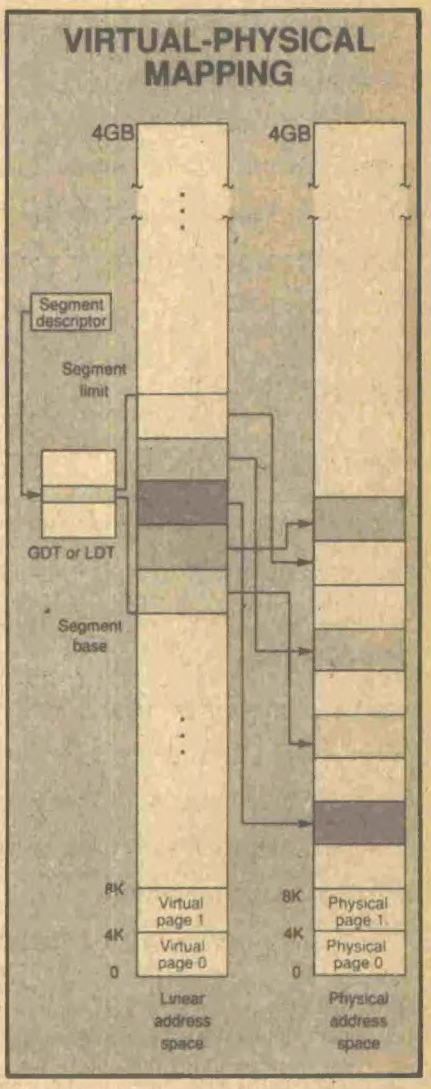
ในการแปลงแอดเดรสเทียมให้เป็นแอดเดรสจริงนั้น เนื่องจากจำเป็นจะต้องอ้างถึงข้อมูลในหน่วยความจำถึง 2 ชุด (สารบาญเพจและตารางเพจ) ซีพียู 80386 และ 80486 จึงมีระบบหน่วยความจำแคชที่ใช้สำหรับจดจำข้อมูลในตารางเพจที่ถูกเรียกใช้ไปแล้วโดยเฉพาะ เรียกว่า TLB (Translation Look-aside Buffer) เพื่อจะได้ไม่ต้องเปิดตารางเพจซ้ำ ซึ่งซอฟต์แวร์ก็อาจยกเลิกการทำงานของ TLB ได้ แต่อย่างไรก็ตาม ซอฟต์แวร์หน่วยความจำเทียมจะต้องยกเลิกข้อมูลใน TLB ทุกครั้งหากมีการเปลี่ยนแปลงของตารางเพจเพื่อให้แน่ใจว่าจะไม่มีการอ้างถึงข้อมูลในตารางเพจเดิมซึ่งได้แก้ไขไปแล้ว

การแปลงแอดเดรสแบบเซกเมนต์และแบบเพจอาจทำงานร่วมกันได้ ดังรูปที่ 5 แสดงให้เห็นเซกเมนต์หนึ่งซึ่งมีขนาด 5 เพจ (20 กิโลไบต์) ทั้ง 5 เพจนี้จะเรียงติดต่อกันแบบลิเนียร์แอดเดรส เมื่อผ่านกระบวนการแปลงแอดเดรสให้เป็นแอดเดรสของเพจจริงแล้วจะได้แอดเดรสของแต่ละเพจจริงที่ไม่ต่อเนื่องและไม่เรียงกัน

ระบบจัดการ Virtual Memory

ดังได้กล่าวมาแล้วว่าหน้าที่ของระบบจัดการหน่วยความจำเทียมก็คือการจัดสรรปันส่วนหน่วยความจำเทียมให้โปรแกรมได้ใช้งาน โดยการโอนย้ายเพจที่ยังไม่ได้ใช้งานจากหน่วยความจำเข้าไปเก็บไว้ในดิสก์ก่อนหากหน่วยความจำเต็ม และโหลดเพจที่ถูกเรียกใช้และยังไม่มีอยู่ในหน่วยความจำจากดิสก์ไฟล์เข้าไปในหน่วยความจำแทน ในส่วนนี้จะเห็นได้ว่าจะต้องมีขั้นตอนในการพิจารณาว่าเพจไหนในหน่วยความจำจะต้องโอนย้ายเข้าไปเก็บในดิสก์ และเพจจริงอันไหนที่จะส่งเข้าไปเก็บในหน่วยความจำเมื่อเพจเทียมของมันถูกเรียกใช้ ซึ่งขั้นตอนนี้โปรแกรมอาจดำเนินการเองได้ ลอจิกในการคัดเลือกเพจที่จะต้องย้ายเข้าไปเก็บในดิสก์มักใช้กัน 3 วิธี ซึ่งคงจะไม่มีวิธีไหนที่ดีที่สุด เพราะทุกวิธีต่างมีโอกาสที่จะตัดสินใจอย่างไม่เหมาะสมได้ เนื่องจากเราไม่รู้ล่วงหน้าว่าเพจไหนที่จะถูกเรียกใช้ต่อไป (อาจจะเป็นเพจที่เพิ่งย้ายไปเก็บในดิสก์ก็ได้) แต่อย่างไรก็ตาม วิธีทั้ง 3 นี้ได้ผ่านการพิจารณามาแล้วว่าเป็นวิธีที่มีเหตุผลสมควรที่สุด ได้แก่

- LRU (Least-Recently Used) คือพิจารณาเฉพาะช่วงเวลาที่เพิ่งผ่านไปว่าเพจไหนในหน่วยความจำที่โปรแกรมเรียกใช้น้อยที่สุด เพจนั้นจะถูกย้ายไปเก็บในดิสก์ โดยวิธีนี้จะต้องมีการประทับเวลาลงในตารางเพจทุกครั้งที่เพจถูกเรียกใช้ วิธีนี้เป็นวิธีที่นิยมใช้กันมากที่สุด
- LFU (Least-Frequently Used) คือพิจารณาจากช่วงเวลาที่ผ่านมาทั้งหมด



รูป 5 แสดงการทำงานร่วมกันของการแปลงแอดเดรสแบบเซกเมนต์และแบบเพจ

ว่าเพจไหนในหน่วยความจำถูกเรียกใช้น้อยที่สุด จะย้ายเพจนั้นไปเก็บในดิสค์ วิธีนี้จะต้องมีการนับจำนวนครั้งที่แต่ละเพจถูกเรียกใช้เอาไว้ในตารางเพจ

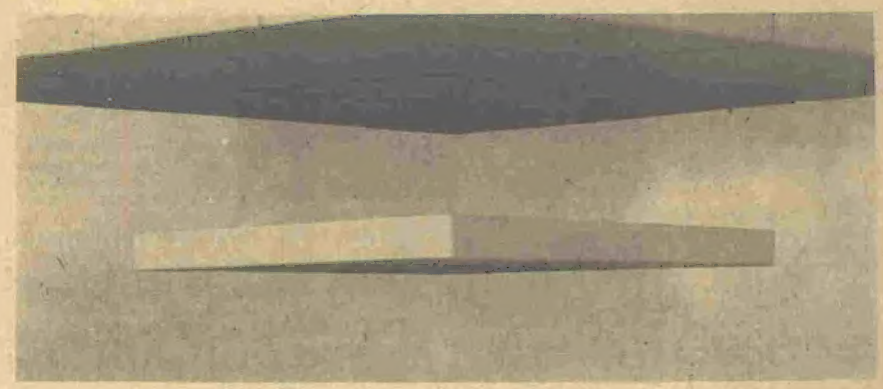
- NUR (Not-Used-Recently) การพิจารณาเฉพาะช่วงเวลาที่เพิ่งผ่านไปว่าเพจไหนในหน่วยความจำไม่ได้เรียกใช้เป็นเวลานาน เพจนั้นจะถูกย้ายไปเก็บในดิสค์ วิธีนี้จะต้องมีการเปลี่ยนเวลาที่ถูกเรียกใช้ครั้งสุดท้ายในตารางเพจทุกครั้งที่เพจถูกเรียกใช้

ระบบจัดการย่อยหน่วยความจำเทียมจะต้องยังคงรักษาโครงสร้างของข้อมูลต่าง ๆ เอาไว้เสมอ ได้แก่ ข้อมูลใน GDT, LDT และข้อมูลในตารางเพจ เพื่อที่โปรเซสเซอร์จะสามารถแปลงแอดเดรสเทียมให้เป็นแอดเดรสจริงได้ อีกทั้งยังต้องรักษาข้อมูลเพิ่มเติมอื่น ๆ ไว้สำหรับการทำงานของหน่วยความจำเทียมเองอีกด้วย

โครงสร้างข้อมูลของระบบ VM

ระบบจัดการย่อยของหน่วยความจำเทียมจะต้องเก็บรักษาข้อมูลตำแหน่งของหน่วยความจำทั้งที่ถูกใช้ไปแล้วและยังไม่ได้นำมาใช้ จากการที่เพจแต่ละเพจต่างมีขนาดเท่ากัน และไม่จำเป็นที่จะต้องอยู่ต่อเนื่องกันจริง ๆ (เนื่องจากสามารถมองให้ต่อเนื่องกันได้ในรูปแบบลิเนียร์แอดเดรส) เพจจริงจึงมีรูปแบบของโครงสร้างเป็นลิงก์ลิสต์

เมื่อเซกเมนต์หนึ่ง ๆ ถูกจับจองหรือขยายขนาดออกไป จะใช้เนื้อที่ที่ต่อเนื่องกันไปในแบบลิเนียร์แอดเดรส ระบบจัดการหน่วยความจำเทียมจึงต้องจดจำตำแหน่งของเนื้อที่ที่ถูกจับจองไปแล้วหรือยังว่างอยู่เอาไว้โดยอาศัย



บิตแมพ ซึ่งแต่ละบิตในบิตแมพจะใช้แทนแต่ละเพจในแบบลิเนียร์แอดเดรส นอกจากนี้ระบบจัดการหน่วยความจำเทียมยังต้องเก็บรักษาข้อมูลเกี่ยวกับแต่ละเพจทั้งที่อยู่ในหน่วยความจำและไม่อยู่ในหน่วยความจำเอาไว้ในตารางเพจด้วย

บทสรุป

ข้อได้เปรียบที่สำคัญของระบบ demand-paged VM ก็คือ

- ความสามารถในการรันโปรแกรมใหญ่ ๆ ได้โดยไม่ต้องคำนึงถึงขนาดของหน่วยความจำที่มีอยู่จริง
- ความสามารถในการที่จะนำโปรแกรมเข้าไปเก็บ ณ ตำแหน่งใด ๆ ในหน่วยความจำก็ได้ ไม่จำเป็นจะต้องอยู่ต่อเนื่องกันไปเหมือนปกติ โดยไม่เกิดผลกระทบต่อการทำงานของโปรแกรม
- การแบ่งโปรแกรมออกเป็นเพจซึ่งมีขนาดแน่นอนทำให้หมดปัญหาเรื่องเศษเหลือของการแบ่งโปรแกรมเป็นส่วน ๆ ที่ไม่ลงตัว
- ความสามารถในการเก็บโปรแกรมอย่างไม่ต่อเนื่องโดยที่โปรแกรมยังมองเห็นว่าต่อเนื่องอยู่

ข้อจำกัดของระบบหน่วยความจำเทียมมีดังนี้

- จำเป็นต้องมีที่ว่างบนแผ่นดิสค์เพียงพอสำหรับโปรแกรมส่วนที่ไม่พอจะเก็บในหน่วยความจำ
- ทำให้โปรแกรมรันได้ช้าลง เนื่องจากการโอนย้ายส่วนของโปรแกรมระหว่างดิสค์กับหน่วยความจำ แต่คงจะไม่ลืมว่าโปรแกรมที่มีขนาดใหญ่กว่าหน่วยความจำเท่านั้นที่จำเป็นจะต้องมีการโอนย้าย ซึ่งหากไม่มีหน่วยความจำเทียมแล้ว โปรแกรมนั้นย่อมจะไม่สามารถทำงานบนคอมพิวเตอร์เครื่องนั้นได้เลย

ด้วยระบบฮาร์ดแวร์ที่สนับสนุนการใช้งานระบบ demand-paged VM ของซีพียู 80386 และ 80486 ทำให้ความสามารถต่าง ๆ ที่ได้กล่าวไปแล้วข้างต้นทั้งหมดนั้นมีใช้กันในระดับไมโครคอมพิวเตอร์ ระบบ demand-paged VM ในปัจจุบันจะใช้ได้กับระบบจัดการยูนิกซ์หรือโดยการใช้โปรแกรมเสริมการทำงานของดอส จากการพัฒนานี้ทำให้ไมโครคอมพิวเตอร์มีความสามารถในการทำงานในระดับที่ทัดเทียมกับคอมพิวเตอร์ตระกูลเวอร์กสเตชันและมินิคอมพิวเตอร์เลยทีเดียว